การจัดการไฟล์ด้วย เทอร์โบปาสคาล 5.0

อาจหาญ สัตยารักษ์

ในการเขียนโปรแกรมด้วยภาษาปาสคาลนั้น บางครั้งเราก็มีความจำเป็นที่ต้องการจัดการเกี่ยวกับไฟล์ในแผ่นบันทึกอยู่บ้าง เช่น อยากจะเปลี่ยนชื่อไฟล์ ลบไฟล์ หรือทำการล็อคไฟล์ไว้ไม่ให้ลบ ฯลฯ ซึ่งการที่ออกไปสู่ดอสแล้วทำการในเรื่องที่ต้องการนั้น ๆ เราก็ทำได้

แต่หากเราอยากจะให้ทำงานดังกล่าวได้ โดยยังคงทำงานอยู่ในปาสคาลอยู่เราก็สามารถทำได้ ทั้งนี้เพราะปาสคาลทั้ง 4.0 และ 5.0 นั้นมีคำสั่งเพื่อการนี้ให้มาแล้ว

คำสั่งที่จะใช้ทั้งหมดในครั้งนี้คือ

1. Assign เพื่อกำหนดไฟล์ที่เราต้องการจัดการด้านต่าง ๆ

2. SetFAttr เพื่อกำหนดค่าแอตทริบิวต์ของไฟล์ในแผ่นบันทึก

3. Rename เพื่อใช้เปลี่ยนชื่อของไฟล์เก่าไปเป็นไฟล์ชื่อใหม่

4. Erase เพื่อทำการลบไฟล์ที่ไม่ต้องการออกไปจากแผ่นบันทึก

5. FindFirst หาชื่อไฟล์แรกในการค้นหาไฟล์จากแผ่นบันทึก

6. FineNext หาชื่อไฟล์ถัดไปในการค้นหาไฟล์จากแผ่นบันทึก

7. UnpackTime ทำการเปลี่ยนค่าเวลาที่อ่านได้จากการค้นหาไฟล์ด้วยคำสั่งข้อ 5-6 ให้เป็นค่าของ วันเดือนปี เวลา ครั้งหลังสุดที่ไฟล์ถูกบันทึกลงแผ่น

นอกจากนี้จะมีตัวแปรที่เกี่ยวข้องอีกมากมาย ซึ่งตัวแปรเหล่านี้ถูกกำหนดมาในระบบของปาสคาลเองแล้ว เราจึงไม่ต้องไปกำหนดตัวแปรเหล่านี้อีก แต่สามารถอ้างถึงนำมาใช้งานได้เลย

ผู้เขียนมีซอฟต์เมล์ประกอบบทความนี้ด้วย ผู้อ่านที่ไม่ประสงค์จะนั่งคีย์โปรแกรมก็ติดต่อกับวารสารได้เช่นเคย

กำหนดไฟล์ต้นแบบก่อน

เพื่อสะดวกในการใช้งานในไฟล์หลัง ๆ ผู้เขียนจึงได้สร้างไฟล์แรกขึ้นมาก่อนโดยมุ่งกำหนดตัวแปรที่จะใช้งานทั้งหมดไว้ที่ไฟล์นี้ และเรียกมาใช้งานด้วยคำสั่ง {$I} ในภายหลัง โดยขอเก็บในชื่อ FILEHEAD (FILEHEAD.PAS ซึ่งเป็นชื่อเดียวกันกับในแผ่นซอฟต์เมล์ด้วย)

โปรแกรมในรูปที่ 1 คือโปรแกรมที่กล่าวถึง

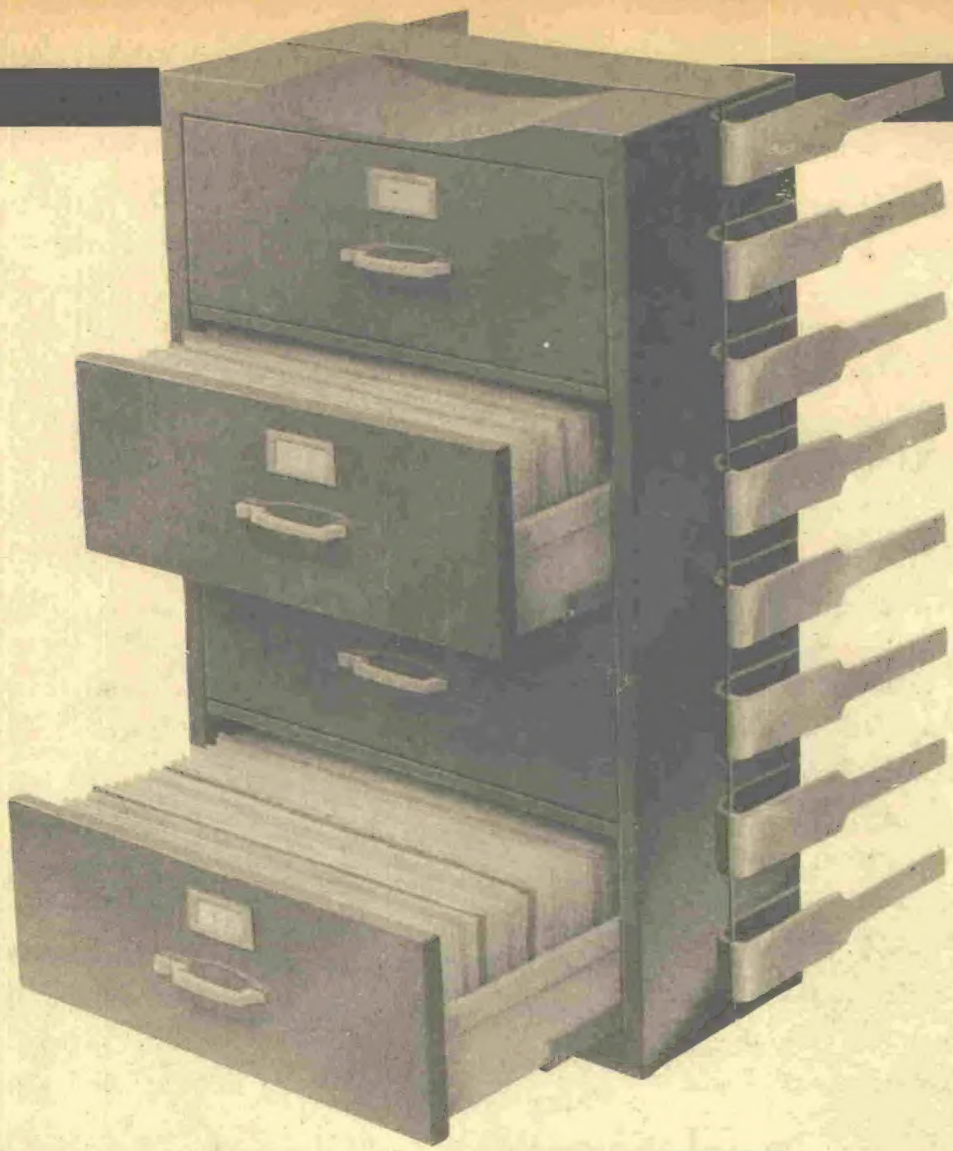
ส่วนตัวแปรที่กำหนดมาทั้งหมดจะนำไปอ้างถึงในโปรแกรมในรูปที่ 2, 3 และ 4 ต่อไปด้วย ดังนี้

Ch,Funckey ตัวแปรอ่านค่าการกดแป้นโดย Ch เป็นอักขระธรรมดา ส่วน Funckey สำหรับอ่านแป้นฟังก์ชันทั้งหมด โดยมีการอ่านโดยตรงจากอินเตอร์รัพต์ที่ควบคุมการกดแป้น ($00) ไม่ใช่การอ่านจากคำสั่ง Readkey ที่ใช้กัน ซึ่งเป็นไปตามชุดคำสั่ง ReadChr; โดยอ่านจากค่า Regs.AH และ Regs.AL ซึ่งเป็นชื่อแบบข้อมูล (Type) มาตรฐานของปาสคาลเองดังรูปที่ 5

เราไม่ต้องกำหนดชื่อแบบข้อมูลดังกล่าวนี้ซ้ำในโปรแกรมแต่อย่างใด ซึ่งจะเหมือนกับชื่อแบบข้อมูลถัดไป คือ SearchRec ซึ่งเป็นข้อมูลเกี่ยวกับไฟล์ เช่น ชื่อไฟล์ วันเดือนปี เวลา ที่ไฟล์ถูกบันทึกไว้

FFile กำหนดไว้เก็บชื่อไฟล์ที่ค้นหามาได้จากคำสั่ง FindFirst และ FindNext ซึ่งได้กำหนดไว้ 80 ชื่อ ส่วนตัวแปรอื่น ๆ คงต้องศึกษาประกอบกับโปรแกรมถัด ๆ ไปอีกที

มาดูชุดคำสั่งที่อยู่ในโปรแกรม



```
USES dos,crt;

VAR Ch,Funckey:char; FileName:string[12];
    DirInfo:SearchRec;
    FFile:array[1..80] of string[12];
    Linedot,LineSp:string[80];
    RowFSt,RowFEnd,ColumnF,NoF,Func,ROW,No:Byte;
    Column:integer;

PROCEDURE Inverse(x,y:integer; Content:string);
begin TextColor(0); TextBackground(7);
   gotoxy(x,y); Write(Content);
   TextColor(3); TextBackground(0);
End;

PROCEDURE Normal(x,y:integer; Content:string);
begin TextColor(3); TextBackground(0);
   gotoxy(x,y); Write(Content);
End;

PROCEDURE ReadChr;
var Regs:Registers;
begin
   regs.AH:=$00; intr($16,regs);
   Func:=regs.AH; ch:=chr(regs.AL); Funckey:=Chr(Func);
end;

PROCEDURE ClearSCR;
begin
     Window(1,1,80,25);
     Normal(1,1,' '); Writeln; ClrScr;
end;
```

รูปที่ 1

โปรแกรม FILEHEAD.PAS

ในรูปที่ 1 ซึ่งมีอยู่ 4 ชุดคำสั่ง คือ

1. ชุดคำสั่ง Inverse ทำหน้าที่แสดงข้อความที่กำหนดให้ (content) ไปแสดง ณ ตำแหน่งที่ระบุ (x,y) โดยจะแสดงเป็นอักษรแบบกลับขาวดำ

2. ชุดคำสั่ง Normal ทำหน้าที่แสดงข้อความที่กำหนดเป็นแบบปกติ

โปรดสังเกตว่าผู้เขียนใช้คำสั่งให้แสดงข้อความโดยมิให้เลื่อนเคอร์เซอร์ลงบรรทัดถัดไป ซึ่งตามคำสั่งนี้เคอร์เซอร์จะคงอยู่ ณ ตำแหน่งสุดท้ายที่แสดงข้อความเสร็จสิ้น ทั้งนี้เพราะต้องการนำไปใช้ประโยชน์ในภายหลัง (โปรแกรม SELECTF.PAS)

3. ชุดคำสั่ง ReadChr สำหรับอ่านค่าการกดแป้นพิมพ์ ซึ่งอ่านได้ทันทีทั้งที่เป็นอักขระธรรมดา และอักขระพิเศษ (ch คือค่าอักขระปกติ, Funckey คือค่าอักขระพิเศษ เช่น F1-F10, Home, PgUp ฯลฯ)

4. ชุดคำสั่ง ClearSCR สำหรับลบจอภาพ ตรงนี้ผู้อ่านคงแปลกใจว่าทำไมจึงไม่ใช้คำสั่ง ClrScr ที่มีอยู่แล้วเลย เหตุผลก็คือผู้เขียนต้องการแน่ใจว่าการลบจอภาพนั้นจะต้องลบให้จอมืดจริง ๆ เพราะว่าตอนนี้ได้มีคำสั่งแสดงข้อความกลับขาวดำไว้ด้วย จึงมาเขียนชุดคำสั่งนี้เพื่อให้ลบจอได้มืดสนิทจริง ๆ เท่านั้น และเป็นการลบจอภาพทั้งหมดด้วย สังเกตได้จากการกำหนด Window (1,1,80,25); ก่อนการใช้คำสั่ง ClrScr;

การจัดการ ซ่อนไฟล์, ล็อคไฟล์,เปลี่ยนชื่อไฟล์ และลบไฟล์

ตอนนี้เราจะสร้างชุดคำสั่งสัก 5 ชุดคำสั่ง เพื่อทำงานเกี่ยวกับไฟล์ในแผ่นบันทึก ผู้อ่านจะได้นำไปใช้งานเมื่อมีความต้องการจัดการบางประการกับไฟล์โดยไม่ต้องออกไปจากปาสคาล อีกทั้งยังสามารถทำให้ไฟล์ที่ถูกซ่อนไว้หรือถูกล็อคไว้(ไฟล์ที่ถูกล็อคไว้จะลบ

ไม่ได้) กลับมาเป็นไฟล์ปกติได้อีกครั้งหนึ่ง

โปรแกรมในรูปที่ 2 คือการทำงานทั้งหมดที่กล่าวถึง (ในซอฟต์เมล์ไฟล์ชื่อ SETFILE.PAS) โดยทุกชุดคำสั่งมีคำสั่ง Assign เพื่อกำหนดไฟล์ที่ต้องการกระทำในแต่ละรายการ โดยรูปแบบคือ Assign(f; name:string); ในที่นี้ผู้เขียนกำหนด FileH ขึ้นมาแทน f และส่งชื่อไฟล์ที่จะจัดการ (FileName หรือ Oldname) ไปแทนที่ name เพื่อให้ปาสคาลรับรู้ไฟล์ที่จะจัดการ ซึ่งเห็นได้จากคำสั่งทั้ง 5 ชุดคำสั่งจะใช้กำหนดในบรรทัดแรก ทั้ง 5 ชุดคำสั่งนี้อาศัยค่าคงที่ของปาสคาลคือ

```
ReadOnly   = $01;
Hidden     = $02;
SysFile    = $04;
VolumeID   = $08;
Directory  = $10;
Archive    = $20;
AnyFile    = $3F;
```

โดยในที่นี้จะนำไปใช้เป็นบางค่าเท่านั้น ที่เหลือก็ให้ผู้อ่านลองศึกษาดูเอง สังเกตว่าเราต้องเริ่มต้นด้วยการเรียกไฟล์ FILEHEAD เข้ามาใช้งานร่วมด้วยในบรรทัดแรก

ชุดคำสั่ง HiddenFile สำหรับจัดการให้ไฟล์ที่กำหนดเป็นไฟล์ชนิดที่ไม่สามารถใช้คำสั่ง dir ในดอสได้ หรือไม่สามารถใช้คำสั่ง FindFirst หรือ FindNext ในปาสคาลเองด้วย โดยใช้คำสั่งกำหนดค่าของแอตทริบิวต์ใหม่ด้วยคำสั่ง SetAtrr(FileH,$02); หรือจะใช้ SetAtrr(FileH,hidden); ก็ได้เช่นกัน

ชุดคำสั่ง LockFile สำหรับทำหน้าที่ล็อคไฟล์ที่กำหนดไว้ เช่น เขียนยังไม่เสร็จเกรงว่าจะมีใครมาเขียนทับก็ให้ล็อคไฟล์ดังกล่าวไว้เสียก่อน โดยกำหนดแอตทริบิวต์ให้เป็น $01

ชุดคำสั่ง ReNameFile เพื่อทำการเปลี่ยนชื่อไฟล์เดิม (OldName) ไปเป็นชื่อไฟล์ใหม่ (NewName) ซึ่งจะทำงานเป็น 2 ขั้นตอนคือกำหนดไฟล์ในชื่อเก่าด้วย Assign(FileH,OldName); และจัดการเปลี่ยนชื่อโดยคำสั่ง ReName (FileH,NewName);

ชุดคำสั่ง EraseFile สำหรับลบไฟล์ที่ไม่ต้องการออกไปจากแผ่นบันทึก สังเกตคำสั่งจะแตกต่างไปเล็กน้อยคือในคำสั่ง Erase นั้นไม่ต้องระบุชื่อไฟล์ลงไปเพราะเราได้กำหนดตอน Assign แล้ว

ชุดคำสั่งสุดท้าย NormalFile เพื่อจัดการให้ไฟล์ที่กำหนดเป็นไฟล์ปกติที่เราจะสามารถเลือกใช้งานได้โดยการกำหนดแอตทริบิวต์ให้เป็น $20;

สำหรับไฟล์ในรูปที่ 2 นี้ก่อนการรันให้สร้างไฟล์ TEST1, TEST2, TEST3 และ TEST4 ไว้ก่อนแล้วจึงรัน(หรือจะลองเปลี่ยนเป็นชื่อไฟล์อื่นดูก็ได้)

ขอให้สังเกตว่าชื่อของไฟล์แรกนั้นเราสามารถกำหนดได้แบบเดียวกับการอยู่ในดอส เช่น

HiddenFile('B:Test1.EXE'); ก็คือการซ่อนไฟล์ชื่อ test1.exe ที่อยู่ในไดรฟ์ B

LockFile('A:*.PAS'); คือการล็อคไฟล์ทุกไฟล์ที่มีสกุลไฟล์เป็น PAS

EraseFile('C:*.*'); ลบไฟล์ทุกไฟล์ในไดรฟ์ C

NormalFile('A:*.PAS'); คือการปลดล็อค หรือปลดการซ่อนไฟล์ .PAS ในไดรฟ์ A ทุกไฟล์

ฯลฯ

ขอให้ท่านลองแก้ไขและรันดูเองจะเห็นประโยชน์ของ 5 ชุดคำสั่งนี้มาก ๆ

```
{$I FILEHEAD }

VAR FileH:file;

PROCEDURE HiddenFile(FileName:string);
begin
  Assign(FileH,Filename);
  SetFAttr(FileH,$02);
  Inverse(WhereX,WhereY,' '+FileName+' is hidden ');
end;

PROCEDURE LockFile(FileName:string);
begin
  Assign(FileH,Filename);
  SetFAttr(FileH,$01);
  Inverse(WhereX,WhereY,' '+FileName+' is lock ');
  Writeln;
end;

PROCEDURE ReNameFile(OldName,NewName:string);
begin
  Assign(FileH,Oldname);
  Rename(FileH,NewName);
  Inverse(WhereX,WhereY,'Rename '+OldName+' to '+NewName+' ');
  Writeln;
end;

PROCEDURE EraseFile(Filename:string);
begin
  Assign(FileH,Filename);
  Erase(FileH);
  Inverse(WhereX,WhereY,' '+FileName+' is erase ');
  Writeln;
end;

PROCEDURE NormalFile(FileName:string);
begin
  Assign(FileH,Filename);
  SetFAttr(FileH,$20);
  Inverse(WhereX,WhereY,' '+FileName+' is normal ');
  Writeln;
end;

{============Program for SetFile in PASCAL 5.0 ====================}
 begin
    ClearSCR;
      HiddenFile('A:TEST1');
      LockFile('A:TEST2');
      ReNameFile('A:Test3','NewFile');
      EraseFile('A:Test4');
      normalFile('A:TEST1');
      normal(WhereX,WhereY,'Press one key');
      ReadChr;
  end.
```

รูปที่ 2 โปรแกรม SETFILE.PAS

การแสดงชื่อไฟล์ที่มีในแผ่นบันทึก

เราคงเคยใช้คำสั่ง dir ในดอสกันมาแล้วอาจสงสัยว่า ดอสสามารถอ่านชื่อไฟล์ จำนวนไบต์ของไฟล์ วันเดือนปี และเวลาที่ไฟล์นั้น ๆ ถูกบันทึกลงไปไว้ ทั้งยังบอกถึงจำนวนไฟล์ที่เราเรียกดู เนื้อที่ในแผ่นบันทึกว่าว่างอยู่อีกเท่าใด โดยคงจะสงสัยว่าดอสอ่านมาได้อย่างไร

เราจะมาลองใช้ภาษาปาสคาลเลียนแบบการค้นหาไฟล์แบบนี้ดูกันหน่อย

คำสั่งที่ใช้คือ FindFirst, FindNext, DiskFree และ UnpackTime โดยใช้ประกอบกันกับตัวแปรและชื่อแบบข้อมูลที่ถูกกำหนดไว้แล้วด้วยปาสคาลดังรูปที่ 6

โดยในที่นี้ผู้เขียนกำหนดให้ DirInfo เป็น SearchRec ไว้ตั้งแต่ไฟล์ FILEHEAD แล้ว(ขอย้อนไปดูในรูปที่ 1 อีกที) พร้อมกับเอาตัวแปรไปใช้ 3 ตัวคือ

Time เพื่อมาหา วันเดือนปี และเวลา จากคำสั่ง UnpackTime โดยค่าตัวแปรนี้ตอนที่นำมาใช้งานจริงก็คือตัวแปรที่มีชื่อว่า DirInfo.Time นั่นเอง

Size เพื่อนำมาหาจำนวนไบต์ของไฟล์ที่อ่านมาได้ โดยปกติค่านี้มีค่าสูงสุดสำหรับแผ่นบันทึกตามปกติเป็นเลข 6 หลัก(แผ่นบันทึกขนาด 360 KB) ตัวแปรที่เอาไปใช้งานจริงคือ DirInfo.Size

Name นี่คือชื่อไฟล์ที่อ่านได้โดย

ปาสคาลจะตรวจสอบให้เองว่าตรงกับที่กำหนดให้อ่านหรือไม่ เช่น ถ้าเราตรวจไฟล์ *.PAS เมื่อเจอไฟล์ชื่อ T.EXE ปาสคาลจะให้ค่า DirInfo.Name มีค่าเป็นไฟล์ว่างทุกไฟล์ไป ดังนั้นเมื่อกำหนดให้ค่า DirInfo.Name เปลี่ยนไปเป็น FFile[No] นั้น เมื่อ FFile[1]:=' '; จึงถือว่าอ่านไฟล์ที่กำหนดไม่พบก็ไม่แสดงชื่อไฟล์บนจอภาพ

มาดูโปรแกรมในรูปที่ 3 (ชื่อไฟล์ในซอฟต์เมล์คือ DIRFILE.PAS) กันต่อ เช่นเดิมเราจะเริ่มด้วยการใช้คำสั่ง {$I FILEHEAD}

มาดูชุดคำสั่งแรกคือ Date_Time เป็นชุดคำสั่งเพื่อหาวันเดือนปีออกมาตามรูปแบบของการใช้คำสั่ง dir ในดอส คือ 1-01-80, 12-31-89 ฯลฯ และจัดรูปแบบของเวลาเช่น 12:01a, 3:25p ฯลฯ เพื่อที่จะแสดงตามหลังชื่อไฟล์นั้น ๆ

คำสั่งที่ใช้คือ UnpackTime ซึ่งเกี่ยวข้องกับชื่อแบบข้อมูล dateTime ดังรูปที่ 7 ซึ่งถึงไม่อธิบายคงรู้ว่าแต่ละตัวแปรหมายถึงอะไร(ผู้เขียนกำหนดในไฟล์เป็น Date) เช่น

Date.Year คือปี ค.ศ. แต่เวลาแสดงบนจอนั้นแสดงเพียง 2 หลักหลัง ดังนั้นในโปรแกรมจึงต้องเอามาเฉพาะ 2 ตัวหลัง (บรรทัดที่ 6 หลังคำสั่ง begin) ซึ่งเป็นอักขระขนาด 2 ไบต์ เช่นปี ค.ศ.

```
{$I FILEHEAD}

VAR  DateF:string[8]; TimeF:String[6];  FileCount:Byte;

PROCEDURE Date_Time(Time:longInt);
var Date:dateTime;  St:string[4];   AmPm:string[1];
begin
  AmPm:='a';            UnpackTime(Time,Date);
  if Date.Hour>=12 then begin  AmPm:='p';  Dec(Date.Hour,12); end;
  if Date.Hour=0 then Date.Hour:=12;
  str(Date.Month,st); If Length(st)=1 then st:=' '+St; DateF:=st+'-';
  str(Date.Day,st);   If Length(st)=1 then st:='0'+St; DateF:=DateF+st+'-';
  str(Date.Year,st);  DateF:=DateF+copy(st,3,2);
  str(Date.Hour,st);  If Length(st)=1 then st:=' '+St; TimeF:=st+':';
  str(Date.Min,st);   If Length(st)=1 then st:='0'+St; TimeF:=TimeF+st+AmPm;
end;

PROCEDURE DirFile(RowSt:byte; FindFile:String);
const Drive:array[1..4] of string[1]=('A','B','C','D');
var   Filename:string[8]; SurName:String[3];
      SizeST,FileCST,ByteFree:string[10];  DriveNo:byte;
      FileCode:byte; Point,LL:byte;
 begin
    DriveNo:=1; FileCount:=0;
    If FindFile[2]=':' then DriveNo:=Ord(Upcase(FindFile[1]))-64;
    FindFirst(FindFile,$20,DirInfo); Gotoxy(1,RoWst);
    Writeln(' Volume in Drive ',Drive[DriveNo],' is ');
    WriteLn(' Directory of  ',Drive[DriveNo],':\');   Writeln;
  Repeat
     Filename:=DirInfo.Name; SurName:='   '; FileCode:=0;  Point:=0;
     Point:=Pos('.',DirInfo.Name);  LL:=Length(DirInfo.Name)-Point;
     if Point<>0 then                          begin
       FileName:=Copy(DirInfo.Name,1,Point-1);
       SurName:=Copy(DirInfo.Name,Point+1,LL);
       Surname:=Copy(SurName+'   ',1,3);       end;
     Filename:=FileName+Copy(LineSp,1,8-Length(FileName));
     Str(DirInfo.Size,SizeST);
     SizeSt:=Copy(LineSp,1,6-Length(SizeST))+SizeST;
     Date_Time(DirInfo.Time);
     WriteLn(Filename,' ',SurName,'   ',SizeST,'  ',DateF,'  ',TimeF);
     Inc(FileCount);      FindNext(DirInfo);
 Until (DosError<>0) ;
     Str(FileCount,FileCSt);
     FileCSt:=Copy(LineSp,1,9-Length(FileCSt))+FileCSt;
     Str(DiskFree(DriveNo),ByteFree);
     ByteFree:=Copy(LineSp,1,7-Length(ByteFree))+ByteFree;
     writeLn(FileCSt,' File(S)   ',ByteFree,' bytes free');
     Writeln; Write(Drive[DriveNo],'>');
end;

{===========Program for Directory show date & Time ================}
  begin
     ClearSCR;              FillChar(LineSp,12,' ');
     DirFile(2,'A:*.*');    ReadChr;
  end.
```

รูปที่ 3 โปรแกรม DIRFILE.PAS

1989 ก็จะได้ 2 ตัวหลังคือ 89 นั่นเอง

Date.Hour, Date.Min จึงหมายถึงชั่วโมงและนาทีนั่นเอง โดยถ้า Date.Hour มีค่า 0-12 ก็จะได้ตัวแปร AmPm เป็น 'a' แต่ถ้ามากกว่านี้จะได้ AmPm = 'p' แทน

การหาวันเดือนปีจึงต้องอาศัยจาก Date.Month, Date.Day และ Date.Year ตามลำดับ หวังว่าคงทำความเข้าใจกับชุดคำสั่งนี้เพิ่มเติมเองได้

ชุดคำสั่งที่ 2 คือ DirFile จะแสดงผลการใช้คำสั่ง dir ในปาสคาลได้คล้าย ๆ กับในดอสเหมือนกัน โดยจะเริ่มที่ตรวจชื่อ FindFile ว่ามีการระบุไดรฟ์หรือไม่ (ค่า FindFile[2]:= ':') เพื่อจะแสดงชื่อไดรฟ์ให้ถูกต้อง

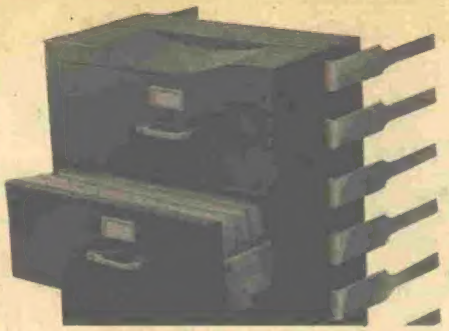
คำสั่ง FindFirst(FindFile,$20, DirInfo); ก็เพื่อหาชื่อไฟล์ วันเดือนปี เวลา ตามที่กล่าวมาแล้วของไฟล์แรก ส่วน FindNext จะหาไฟล์ถัดไปจนหมดไฟล์ในแผ่นบันทึก

จากนี้ทำการตรวจแยกชื่อไฟล์ ชื่อสกุลของไฟล์ออกจากกัน โดยตรวจสอบจากตัว . (จุด) ที่มีในชื่อไฟล์ (ถ้าไม่มีจุดทศนิยมก็แสดงว่าไฟล์นั้นไม่มีชื่อสกุล ก็ต้องกำหนด Surname:= ' ';)

ตรวจสอบวันเดือนปีโดยเรียกชุดคำสั่งแรก คือ Date_Time(DirInfo.Time); แล้วก็เขียนชื่อลงในบรรทัดที่ต้องการตามลำดับคือ ชื่อไฟล์ สกุลไฟล์ จำนวนไบต์ วันเดือนปี เวลา

เมื่อไฟล์ถูกอ่านแต่ละครั้ง ถ้าตรงกับไฟล์ที่ระบุ ค่าของตัวแปร FileCount จะเพิ่มขึ้นหนึ่งค่าจาก Inc(FileCount); เพื่อนำไปแสดงจำนวนไฟล์ที่อ่านได้

ดูจำนวนเนื้อที่ที่เหลือในแผ่นจากคำสั่ง DiskFree; ก็จะทำให้โปรแกรมในรูปที่ 3 นี้ทำงานได้คล้ายกับการ dir ในดอสเช่นกัน ผู้อ่านสามารถใช้คำสั่งกำหนด ชื่อสกุล ชื่อไดรฟ์ได้ตามปกติ ซึ่งขอให้ลองดูเอง การทำงานของโปรแกรมผู้อ่านต้องศึกษาจากจอภาพเอาเอง



เลือกไฟล์จากแผ่นบันทึกโดยไม่ต้องพิมพ์ชื่อ

ผู้อ่านที่ใช้ปาสคาลกันมาคงจะคุ้นกับกรรมวิธีในการเลือกไฟล์ได้ดีกว่าเมื่อเราต้องการถ่ายทอดไฟล์ที่ต้องการมาสู่หน่วยความจำก็เพียงแต่กดแป้น F3 จากนั้นก็มีตารางชื่อไฟล์ให้เราเลือก โดยการเลื่อนแถบสว่างด้วยการใช้แป้นลูกศร 4 แป้นไปยังชื่อไฟล์ที่ต้องการแล้วกดแป้น Enter

คราวนี้เราจะเขียนโปรแกรมในทำนองเดียวกันดูบ้าง โดยอาศัยหลักการที่ว่า

1. จอภาพประกอบด้วย 25 แถว แถวละ 80 สดมภ์

2. ชื่อไฟล์ยาวชื่อละ 12 สดมภ์ สูงสุด คือ ชื่อ 8 สดมภ์, ชื่อสกุล 3 สดมภ์ เช่น FILEHEAD.PAS

3. เว้นช่องห่างระหว่างชื่อ 2 สดมภ์ ดังนั้น 1 ไฟล์จะใช้ 14 สดมภ์

4. ผู้เขียนจะแสดงแถวละ 5 ไฟล์ จึงใช้ 14×5 = 70 สดมภ์ โดยตีเส้นกรอบซ้ายสุดที่สดมภ์ที่ 5 และแสดงชื่อไฟล์แรกที่สดมภ์ที่ 7 (Column:=7) ก็สามารถตีกรอบด้านขวาได้ที่สดมภ์ที่ 76 รหัสที่ใช้ตีกรอบนี้ ผู้เขียนใช้รหัสตามของราชวิถี WORDPC ดังรูปที่ 8 หากเครื่องของผู้อ่านไม่ตรงตามนี้ก็ให้เปลี่ยนไปใช้รหัสอื่น

5. ผู้เขียนกำหนดไฟล์ไว้สูงสุด 80 ไฟล์จึงจะใช้ 16 บรรทัดสำหรับแสดงชื่อไฟล์ได้สูงสุด รวมกับที่ตีกรอบบนและล่างรวมใช้ทั้งสิ้น 18 บรรทัด (แต่ไม่ได้ตีกรอบเต็ม 18 บรรทัดทุกครั้ง เพราะจะตีกรอบล่างเมื่อหมดชื่อไฟล์เท่านั้น)

โปรแกรมการทำงานเป็นดังรูปที่ 4 (ผู้เขียนใช้ชื่อในซอฟต์เมล์ว่า SELECTF.PAS)

ใช้คำสั่ง {$I FILEHEAD} เพื่อเรียกไฟล์นี้มาทำงานร่วมด้วย

ลองมาดูชุดคำสั่งแรกก่อน คือ Dir(RowSt:byte; FindFile:string); ซึ่งเป็นส่วนที่จะนำชื่อไฟล์ในแผ่นบันทึกมาแสดงในกรอบที่สร้างไว้ทีละบรรทัด แบบกลับขาวดำ ตัวแปร RowSt คือแถวแรกที่ต้องการตีกรอบ ชื่อไฟล์จะเริ่มในแถวถัดไปจาก Inc(Row) ตัวแปร FindFile คือส่วนที่เราต้องการให้แสดงชื่อไฟล์ในกรอบ เช่น

A:*.* แสดงชื่อไฟล์ทุกไฟล์ที่ตรวจพบ

A:* แสดงชื่อไฟล์เฉพาะที่ไม่มีชื่อสกุลเช่นไฟล์ของราชวิถี WORD PC เป็นต้น

B:*.PAS แสดงชื่อไฟล์ในไดรฟ์ B เฉพาะที่มีชื่อสกุลเป็น PAS

ฯลฯ

ดังนั้นต้องการให้แสดงชื่อไฟล์ใดจากไดรฟ์ใด ก็ให้กำหนดเอาเองได้ตามต้องการ

เริ่มด้วยกำหนดให้ FFile[1..80] มีค่าเริ่มต้นก่อน โดยต้องการเอาไปตรวจสอบว่าได้อ่านไฟล์มาตามต้องการหรือยัง คือถ้า FFile[1]:='' ก็ให้บอกว่าไม่พบไฟล์ที่ต้องการ

ใช้คำสั่ง FindFirst หาชื่อไฟล์แรกแต่คราวนี้เราไม่จำเป็นต้องตรวจวันเวลา แล้วทำการตีกรอบบนสุดที่ RowSt โดยเป็นแบบกลับขาวดำจากคำสั่ง Inverse ที่กำหนดไว้ในไฟล์ FILEHEAD.PAS

ในช่วง repeat ก็เริ่มด้วยการแสดงชื่อไฟล์แรก ณ ตำแหน่งที่คำนวณได้ (Column,Row) แล้วขยับค่า column เพิ่มอีก 14 สดมภ์จาก Inc(Column, 14) ใช้คำสั่ง FindNext ตรวจหาชื่อไฟล์ถัดไปถ้าเจอก็ไปแสดงชื่อไฟล์ ถ้า

```
{$I .FILEHEAD}

PROCEDURE Dir(RowSt:byte; FindFile:String);
 begin
   For No:=1 to 80 do FFile[No]:='';
   Column:=7; Row:=RowSt; RowFSt:=Row+1; No:=1;
   FindFirst(FindFile,$20,DirInfo);
   Inverse(5,Row,#149+copy(Linedot,1,70)+#151); Inc(Row);
   Inverse(5,Row,#159+Copy(lineSP,1,70)+#159);  RowFEnd:=Row;
  Repeat
     FFile[no]:=DirInfo.name;
     if FFile[no]<>'' then begin
     Inverse(Column,Row,Copy(Dirinfo.name+Copy(LineSP,1,12),1,12));
     NoF:=No; Inc(no);  ColumnF:=Column;  RowFEnd:=Row; Inc(Column,14);
                        end;
     FindNext(DirInfo);
     if Column>63 then begin column:=7;
        Inc(Row); Inverse(5,Row,#159+Copy(lineSP,1,70)+#159);  end;
 Until (DosError<>0) ;
     Inverse(5,RowFend+1,#155+copy(Linedot,1,15)+' arrow key to move  :'+
       '  F1 to Select File'+copy(Linedot,1,15)+#157);
   if FFile[1]='' then Inverse(33,RowSt+1,'File not Found');
end;

PROCEDURE  SelectFile;
begin   No:=1;   Column:=7; Row:=RowFSt;
  repeat
    Normal(Column,Row,FFile[no]);   Readchr;
    Inverse(Column,Row,FFile[no]);
    case Funckey of
      #72:begin if Row>RowFSt  then begin Dec(Row); Dec(no,5); end;
          end;
      #75:begin Dec(Column,14); Dec(no);
                if (Column<7) and (Row>RowFSt) then Begin
                           Dec(Row); Column:=63;   end;
          end;
      #77:begin if Column<65 then begin Inc(Column,14); Inc(no); end;
                if Column>63 then begin Inc(Row); Column:=7; end;
          end;
      #80:begin if Row<RowFEnd then begin Inc(Row); Inc(no,5); end;
          end;
      #71: begin No:=1; Column:=7; Row:=RowFst; end;
      #79: begin No:=NoF; Column:=ColumnF;  Row:=RowFEnd; end;
    end; {End case }
     if No>NoF then begin no:=noF; Row:=RowFEnd; Column:=ColumnF; end;
     If No<1 then begin No:=1; Row:=RowFSt; Column:=7; end;
  until FunCkey=#59;
    FileName:=FFile[no];
end;

{================Program Directory and Select File  ======================}
begin
  ClearSCR;
  Fillchar(Linedot,80,#158); FillChar(Linesp,80,#255);
  Dir(2,'B:*.*');
  if FFile[1]<>'' then begin SelectFile;
             Normal(1,23,'Select File :'+Filename);  end;
 ReadChr;
 end.
```

รูปที่ 4 โปรแกรม SELECTF.PAS

ไม่เจอ (DosError<>0) ก็ให้ออกไปตีกรอบล่างที่แถวที่ RowFend+1 (RowFST คือแถวบนสุดที่แสดงชื่อไฟล์)

ในกรอบข้างล่างนี้จะมีข้อความว่า arrow key to move : F1 to Select File ให้ทราบถึงแป้นพิมพ์ที่จะใช้งานต่อไป

ขอให้สังเกตว่าตัวแปรส่วนมากนั้นกำหนดไว้ตั้งแต่ไฟล์ FILEHEAD แล้ว และอย่าลืมว่าผู้เขียนไม่ได้ตรวจสอบความผิดพลาดหรือหยุดอ่านไฟล์ในกรณีที่มีไฟล์ในแผ่นบันทึกมากกว่า 80 ไฟล์ เพราะต้องการทิ้งไว้ให้ผู้อ่านคิดเองบ้าง

คราวนี้มาถึงตอนทำการเลือกไฟล์กันแล้ว ชุดคำสั่งที่ช่วยงานนี้คือ SelectFile โดยการใช้แป้นลูกศรเลื่อนซ้าย ขวา ขึ้น ลง เราก็จะเลื่อนแถบแสดงไฟล์ (เป็นแถบปกติในกรอบกลับขาวดำของกลุ่มไฟล์อื่น ๆ) ไปยังไฟล์ที่เราต้องการได้

ขณะนี้ลักษณะบนจอภาพจะคล้ายกับที่เห็นในปาสคาล เพียงแต่กรอบจะกว้างกว่าเท่านั้น (หากอยากได้กรอบผิด

```
Type Registers = Record
                 case interger of
                   0: (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags :word);
                   1: (AL,AH,BL,BH,CL,CH,DL,DH :Byte);
                 end;
```

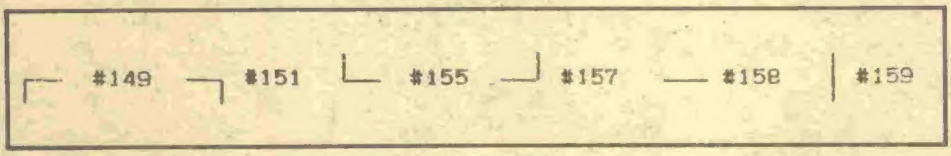
รูปที่ 5

```
Type SearchRec = Record
                 Fill: array[1..21] of byte;
                 Attr: byte;
                 Time: Longint;
                 Size: Longint;
                 Name: String[12];
                end;
```

รูปที่ 6

```
Type DateTime = Record
                Year, Month, Day, Hour, Min, Sec : word;
                end;
```

รูปที่ 7



รูปที่ 8

ไปจากนี้ก็ขอให้ผู้อ่านลองกำหนดเอาเอง)

โดยการตรวจการกดแป้นจากชุดคำสั่ง ReadChr; โดยตรวจจากค่า Funckey ได้ดังนี้

#72 คือการกดแป้นลูกศรขึ้น จะเลื่อนไฟล์ถอยหลังไป 5 ไฟล์ จากคำสั่ง Dec(No,5)

#80 คือการกดแป้นลูกศรลง จะเลื่อนไฟล์ไปข้างหน้า 5 ไฟล์ จากคำสั่ง Inc(No,5)

#75 คือการกดแป้นลูกศรซ้าย จะเลื่อนไฟล์ถอยหลังไป 1 ไฟล์ จากคำสั่ง Dec(No)

#77 คือการกดแป้นลูกศรขวา จะเลื่อนไฟล์ไปข้างหน้า 1 ไฟล์ จากคำสั่ง Inc(No)

#71 คือการกดแป้น Home จะตรงไปยังไฟล์แรก

#79 คือการกดแป้น End จะตรงไปยังไฟล์สุดท้าย

เมื่อได้ไฟล์ที่ต้องการก็ให้กดแป้น F1 (ถ้าต้องการใช้แป้น Enter เป็นแป้นเลือกไฟล์ก็ให้เปลี่ยนที่คำสั่ง until คือ เปลี่ยนไปเป็น until ch = #13;) ซึ่งจะได้ชื่อไฟล์ที่เลือกมาตามคำสั่งท้ายสุดของชุดคำสั่งนี้ คือ FileName:=FFile[No]; ซึ่งจะเอาไฟล์นี้ไปจัดการอย่างไรก็ได้ เช่น ลบ, ล็อค, ซ่อน, เรียกไฟล์ ก็ตามแต่ผู้อ่านจะต้องการ ซึ่งส่วนนี้คงต้องทดลองประยุกต์เอาเอง

ในกรณีที่แสดงนี้ผู้เขียนใช้คำสั่ง Dir(2,'B:*.*'); ซึ่งหมายถึงจะตีกรอบตั้งแต่บรรทัดที่ 2 และตรวจไฟล์ทุก ๆ ไฟล์ในไดรฟ์ B

จะสามารถเลือกไฟล์ได้ก็ต่อเมื่อชื่อไฟล์แรกถูกอ่านพบ (FFile[1]<>'') เท่านั้น

หวังว่าผู้อ่านคงมีแนวทางนำไปใช้ประโยชน์ในการเขียนโปรแกรมภาษาปาสคาลของท่านได้ทางหนึ่ง ก่อนจบผู้เขียนขอฝากข่าวมาว่า ขณะนี้ผู้เขียนออกโปรแกรมที่น่าสนใจคือ

1. โปรแกรมพิมพ์อักษรภาษาไทย-ภาษาอังกฤษขนาดใหญ่ 8 ขนาด ใช้กับจอโมโนโครมและเครื่องพิมพ์แบบ 9 เข็ม

2. โปรแกรมวิชาฟิสิกส์ 3 เรื่อง คือ เลนส์ กฎการเคลื่อนที่ของนิวตัน โปรเจกไตล์

3. ต้นแบบภาษาไทยกราฟิก (ปาสคาล 5.0)

4. โปรแกรมอธิบายการใช้ราชวิถี WORD PC ซึ่งเป็นโปรแกรมประเภท CAI หน้าตาของจอภาพจะเหมือนราชวิถีทุกประการ เรียนการใช้ทีละขั้นตอน และเรียนรู้เมนูการพิมพ์ลงเครื่องพิมพ์พร้อมกับไฟล์เพื่อแก้ไขตำแหน่งแป้นพิมพ์ และตั้งค่าตัวแปรในไฟล์ RWINST.PRM และ NLQINST.PRM ให้ด้วย

5. โปรแกรมประกอบหนังสือเทคนิคการเขียนกราฟิกบนเทอร์โบเบสิก

โดยจัดเป็นโปรแกรมละ 1 แผ่น ถ้าสนใจก็สอบถามไปที่ผู้เขียนได้